M1T1L1_Class_Introduction

- **Objective**: to understand why applied natural language processing is necessary and a booming field in artificial intelligence and machine learning field.

1. **why natural language processing is a very important field in machine learning arena?**
    - Language is a basic and primary tools for human beings to communicate and develop, they are everywhere.
    - We have hundreds of languages in the world, and each produces large amounts of textual information.
    - Terabyte of textual information is produced on a daily basis and understanding the concept behind them is immensely important to create meaningful machine learning models.
    - The good news is that many big and small companies are looking for this skill.

    - Texts and written information are ubiquitous, they can be found anywhere in our life.
        - Mainly thanks to the Internet.
        - People generate a lot of text on web pages such as Twitter, Facebook, Wikipedia, blogs, and many other websites.
        - Books used to be in a paper form and now many libraries, such as university libraries, are digitizing books and research papers.
        - We produce text even for videos, for example captions.
        - Other example for texts and documents are found in police case report, legislation, review and products, medical reports, and job description.

    - Now with this great number of texts and documents out there, how natural language processing can help us? (advantages)
        - establishing authorship and plagiarism detection
        - classification, for example, genres for narratives and tone classification, which is a part of sentiment analysis.
        - syntax analysis for programming codes
        - machine translation, for example, Google Translate

    - Bridging between linguistics and machine learning make natural language processing **a challenging topic** in the field of AI.
        - human language comes in different forms and shapes and that creates ambiguity in multiple levels, for example lexical, which is a word-level ambiguity.
        - That a word might **have different meanings**.
          As an example, let's focus on this sentence: I went to the bank.
          Here bank can have two meanings, it can be either a type of financial institution or an area of a land next to a river.
        - The other ambiguity is **syntax**, which we can just parse sentence in different ways.
          For example, Wafa said on Monday she would give an exam. This sentence means either that it was on Monday that Wafa told the class about the exam or that the exam would be given on Monday.

2. What will You Learn in the Class?

- **pre-processing** of words
  - how to clean texts and documents?
  - What is a tokenization?
  - How to reduce the inflectional forms of the words, such as stemming, and lemmatization,
  - and normalization?
- **text representation**
  - Some of the main and well-known text representation are one hot encoding, bag-of-words, TF-IDF, and embeddings.
- conventional **machine learning** algorithms in the area of natural language processing
  - naive Bayes, logistic regression, support vector machine, perceptron, and neural network (supervised learning)
  - supervised learning and what is the difference between supervised and unsupervised learning
- **deep learning** models
  - convolutional neural network, recurrent neural network, and long short-term memory methods
- **unsupervised learning** algorithms as part of topic modeling
  - principal component analysis, singular value decomposition, and Latent Dirichlet allocation
- **Transformers** play a crucial role in natural language processing nowadays.
  - bidirectional encoder representations from transformers
  - generative pre-trained transformers

3. What Deliverables Do We Expect from You?

- Requirements
  - 4 major homework

    Homework 1 - text pre-processing and classification introduction

    Homework 2 - classification methods; dimensionality reduction and singular value decomposition

    Homework 3 - deep learning algorithms

    Homework 4 - transformers and unsupervised models
  - 10 quizzes on a weekly basis

    measure your understanding of the topics, mostly conceptual questions

    multiple-choice questions

    limited time to finish it

    only cover materials taught in the class

Summary

- quickly went over some of the pre-processing methods that we are going to cover in future
- different text representation methods and techniques
- The programming language which is used in this class is part-time
- implement different natural language processing method, such as converting textual information into a numerical information

- Main goals:
  - to explain different NLP methods
  - to develop and assess the performance of different NLP methods using a variety of techniques.

M1T2L1_Text_Preprocessing_Techniques

Today we'll start diving into preprocessing techniques. During this video, we will start with getting familiar with some NLP terminology, as well as practice some basic preprocessing techniques on text data.

- Some **LLP terminology**
  - **corpus** as a collection of text data
    Think about Yelp reviews, for example, Wikipedia, or a set of media articles that we're trying to categorize.
  - Languages are organized in a certain sequence respecting grammatical rules → **syntax**
    In the example shown here, this is a simple sentence. This is a determinant followed by the verb "is" and then another determinant, "a" and an adjective and a noun.
  - **Syntactic parsing** - the analytical process used to break down text data based on the grammatical rules
  - **semantics** as the actual meaning that is carried by your word or text.
  - **Tokenization** - the process of splitting the text data into smaller chunks
    We can perform tokenization in different ways.m
    The example is tokenizing the sentence based on words.
    "This is a simple sentence" becomes thus broken down into individual words or tokens, "This" "is" etc.
  - **vocabulary** - the set of unique words that are present in our dataset
  - **Stop words** - the words that are very common in the language
    However, they don't add a lot of meaning to the sentence, think for example "a" is the et cetera.
  - **N-Grams** - consecutive sequence of n words, common use of n between 2 and 5
    unigrams for single words, **bigrams** meaning two words, **trigrams**, three consecutive words, and so on
    In the example above, we break the sentence "This is a simple sentence" into bigrams to get "this is", "is a" "a simple" and "simple sentence."

- **how work with text data**
  - **text data** is an **unstructured data**, meaning, every instance can have a different number of words that carry each different meaning. Traditional methods used for numerical data might thus not transfer well to text data.
  - **Preprocessing of text data** is the **first step to cleaning and preparing the data** so we can build a model.
    While text preprocessing may vary a lot from a text to another, it may lead to a better model performance in some cases, think for example the case of bag-of-words model. However, there are some instances where it may reduce the model performance, for example, for neural networks that are trained on embeddings so use it wisely.

- **libraries** that are useful for text preprocessing, namely,
  RE for the regular expression operation, and NLTK for the Natural Language Toolkit
- **common steps** in text preprocessing: noise removal, tokenization, and text normalization
  - **Noise removal** aims at removing unwanted text formatting information like punctuation, special characters, numbers, whitespaces, et cetera using the regular expression library to remove noise.
    The output of the code is the sentence without all the special characters, spaces, or numbers.
  - **Tokenization**
    how to use the NLTK to tokenize our sentence, "This is a simple sentence." As you can see here in the output, the word tokenize function broke down our sentence into a list of individual words.
  - **text normalization**, two techniques are used to bring the texts into a more standard form and to focus on the meaning that is carried and reduce variations.
    Stemming and lemmatization are two techniques that aim at reducing the inflectional forms of words to their base form.
    - **stemming** algorithm work by removing suffixes and prefixes from words to reduce them to their base form, which we call stem.
    - **Lemmatization** performs a similar task. However, it uses lexical knowledge bases to get the correct base forms of the words instead of just chopping off inflections. We provide a code snippet here that uses Porter stemmer and word lemmatizer to perform stemming and lemmatization on an example sentence.

During this class, you are introduced to some NLP terminology and exposed to different text preprocessing steps.